#include <stdio.h>

#include <string.h>

// including necessary header files

#define MAX\_C 11

// defining max number of candidates

// Structure of candidate

typedef struct Candidate {

char name[50];

int votes;

char symbol;

} Candidate;

// global array of candidate details

Candidate allCandidates[MAX\_C];

// global variable to keep the cound of candidates

int candidateCount = 0;

// global array to store all symbols

char symbols[10]

= { '!', '@', '#', '$', '%', '^', '&', '\*', '~', '+' };

// array to keep track of taken symbol

int symbolTaken[11];

// function declaration

void fillCandidate(int);

void displayAllCandidates();

void getVotes(int);

void getResults();

// driver code

int main()

{

// initializing necessary data structures

for (int i = 0; i < 11; i++) {

symbolTaken[i] = 0;

}

// getting the number of candidates

printf("Enter the number of candidates: ");

scanf("%d", &candidateCount);

if (candidateCount >= MAX\_C) {

printf("Number of candidates cannot be greater "

"than 10.\n Terminating the program\n\n");

return 0;

}

// filling the details of the candidate

for (int i = 0; i < candidateCount; i++) {

fillCandidate(i);

}

// getting the number of voters

int numVoters;

printf("Enter the number of voters: ");

scanf("%d", &numVoters);

// Collecting votes

for (int i = 0; i < numVoters; i++) {

getVotes(i);

}

// printing results

getResults();

return 0;

}

// function to populate the allCandidates array using the

// details provided by user

void fillCandidate(int cNum)

{

printf("Available Symbols: \n");

for (int j = 0; j < 10; j++) {

if (symbolTaken[j] == 1)

continue;

printf("%d %c\n", j + 1, symbols[j]);

}

int num = 0;

printf("\nEnter the symbol number of candidate %d: ",

cNum + 1);

scanf("%d", &num);

if (num <= 0 || num > 10 || symbolTaken[num - 1] == 1) {

printf("This Symbol is not available. Please "

"choose from the available symbols\n");

num = 0;

fillCandidate(cNum);

}

else {

symbolTaken[num - 1] = 1;

allCandidates[cNum].symbol = symbols[num - 1];

printf("Enter the name of candidate %d: ",

cNum + 1);

scanf("%s", allCandidates[cNum].name);

allCandidates[cNum].votes = 0;

}

}

// function to display all candidates name with symbol

void displayAllCandidates()

{

if (!allCandidates || !candidateCount) {

perror("Invalid Candidate Array\n");

return;

}

for (int j = 0; j < candidateCount; j++) {

printf("%s\t\t", allCandidates[j].name);

}

printf("\n");

for (int j = 0; j < candidateCount; j++) {

printf("%3c\t\t\t", allCandidates[j].symbol);

}

printf("\n");

}

// function to get votes

void getVotes(int voterCount)

{

displayAllCandidates();

printf("Voter %d, please enter your choice (1-%d): ",

voterCount + 1, candidateCount);

int choice;

scanf("%d", &choice);

// checking for valid choice

if (choice >= 1 && choice <= candidateCount) {

allCandidates[choice - 1].votes++;

}

else {

printf("Invalid choice! Please vote again.\n");

getVotes(voterCount);

}

}

// function to get results

void getResults()

{

int maxVotes = 0;

int winnerIndex = -1;

int winnerFrequency = 0;

for (int i = 0; i < candidateCount; i++) {

if (allCandidates[i].votes > maxVotes) {

maxVotes = allCandidates[i].votes;

winnerIndex = i;

}

}

for (int i = 0; i < candidateCount; i++) {

if (allCandidates[i].votes == maxVotes) {

winnerFrequency++;

}

}

printf("\n-----RESULT-----\n");

if (winnerFrequency > 1) {

printf("No candidate has majority votes\n");

}

else if (winnerIndex != -1) {

printf("The winner is: %s\nCandidate Symbol: "

"%c\nwith %d votes!\n",

allCandidates[winnerIndex].name,

allCandidates[winnerIndex].symbol, maxVotes);

}

else {

printf("No winner\n");

}

}